

ADVANCED DOCKER

[PSCHIFFE.GITHUB.IO/ADVANCED-DOCKER](https://pschiffe.github.io/advanced-docker)

Peter Schiffer

container

noun (*pl. containers*) **1** (*Software*) Fancy process.

clone() + namespaces + cgroups + cow fs + capabilities +
selinux + seccomp

file ➡ process

image ➡ container

```
docker run -it --rm fedora bash
```

```
docker run -d --name my-nginx nginx
```

```
docker logs my-nginx
```

```
docker exec -it my-nginx sh
```

```
docker inspect my-nginx
```

```
docker rm -fv my-nginx
```

STORAGE

- container cow storage
- persistent storage

CONTAINER COW STORAGE

- devicemapper loopback
- devicemapper direct lvm
- overlay
- btrfs
- aufs

/etc/sysconfig/docker-storage-setup

```
DEVS=' /dev/sdb '  
VG=' docker -vg '  
DATA_SIZE=20G
```

PERSISTENT STORAGE

```
docker run -d -e MYSQL_ROOT_PASSWORD=pw \  
--name my-maria mariadb:10.1
```

```
docker run -d -v /var/lib/mysql \  
-e MYSQL_ROOT_PASSWORD=pw \  
--name my-maria mariadb:10.1
```

```
$ sudo docker volume ls  
local          15fc72513a67...
```

```
$ sudo ls -lh /var/lib/docker/volumes  
drwxr-xr-x 3 root root 4.0K Oct  7 22:55 15fc72513a67...
```

```
docker run -d -v my-maria-data:/var/lib/mysql:Z \
-e MYSQL_ROOT_PASSWORD=pw \
--name my-maria mariadb:10.1
```

```
$ sudo docker volume ls
local          15fc72513a67...
local          my-maria-data

$ sudo ls -lh /var/lib/docker/volumes
drwxr-xr-x 3 root root 4.0K Oct  7 22:55 15fc72513a67...
drwxr-xr-x 3 root root 4.0K Oct  7 23:17 my-maria-data
```

```
docker run -d -v /mnt/storage/my-maria:/var/lib/mysql:Z \
-e MYSQL_ROOT_PASSWORD=pw \
--name my-maria mariadb:10.1
```

```
docker run -d \
-v /home/petko/html:/usr/share/nginx/html \
--security-opt label:disable --name my-nginx nginx
```

NETWORKING

`docker run -d --name my-nginx nginx`

```
$ curl 172.17.0.2:80
```

`docker run -d -p 8080:80 --name my-nginx nginx`

```
$ curl localhost:8080
```

`docker run -d --net host --name my-nginx nginx`

```
$ curl localhost:80
```

LINKING CONTAINERS: DEFAULT BRIDGE

```
docker run -d -v my-wp-db-data:/var/lib/mysql:Z \  
    -e MYSQL_ROOT_PASSWORD=pw \  
    --name my-wp-db mariadb:10.1
```

```
docker run -d -p 8080:80 -v my-wp-data:/var/www/html:Z \  
    --link my-wp-db:mysql --name my-wp wordpress:4.6
```



```
$ sudo docker exec -it my-wp bash
root@a453ddb7ea8f:/var/www/html# cat /etc/hosts
...
172.17.0.2      mysql ab1de7043c14 my-wp-db
172.17.0.3      a453ddb7ea8f
root@a453ddb7ea8f:/var/www/html# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
root@a453ddb7ea8f:/var/www/html# env | grep MYSQL | sort
MYSQL_ENV_MYSQL_ROOT_PASSWORD=pw
MYSQL_PORT=tcp://172.17.0.2:3306
MYSQL_PORT_3306_TCP_ADDR=172.17.0.2
MYSQL_PORT_3306_TCP_PORT=3306
...
```

MYSQL_ENV_MYSQL_ROOT_PASSWORD
{LINK ALIAS}_ENV_{ENV NAME FROM LINKED CONT.}

LINKING CONTAINERS: CUSTOM NET

```
docker network create my-wp
```

```
docker run -d -v my-wp-db-data:/var/lib/mysql:Z \  
-e MYSQL_ROOT_PASSWORD=pw --net my-wp \  
--net-alias mysql --name my-wp-db mariadb:10.1
```

```
docker run -d -p 8080:80 \  
-e WORDPRESS_DB_PASSWORD=pw \  
-v my-wp-data:/var/www/html:Z --net my-wp \  
--name my-wp wordpress:4.6
```

```
$ sudo docker exec -it my-wp bash
root@08e10bb15cb5:/var/www/html# cat /etc/hosts
...
172.19.0.3      08e10bb15cb5
root@08e10bb15cb5:/var/www/html# cat /etc/resolv.conf
nameserver 127.0.0.11
root@08e10bb15cb5:/var/www/html# env | grep MYSQL | sort
root@08e10bb15cb5:/var/www/html#
```

BUILDING IMAGES

FROM (fedora:24|centos:7|alpine:3.4|scratch)

fedora:24

(204.4 MB) Recent packages, should be your default

centos:7

(196.7 MB) Stable, not many changes, older packages

alpine:3.4

(4.799 MB) Super small, but with pkg manager

scratch

Special case, empty base image

- dnf | yum | apk
- statically linked binary (go)
- layered images
 - FROM fedora:24
 - FROM my-base:prod
 - FROM my-app:prod

MINIMUM NUMBER OF LAYERS

```
RUN mkdir -p /opt/kibana \  
  && curl -sSL https://elastic.co/kibana-4.6.1-linux-x64.tar.gz \  
    | tar -xzc /opt/kibana --strip 1 \  
  && chown -R root: /opt/kibana
```

```
ENV MY_VAR1=xxx \  
  MY_VAR2=yyy \  
  MY_VAR3=zzz \
```

github.com/goldmann/docker-squash

MINIMUM SIZE OF THE IMAGE

```
FROM fedora:24
RUN dnf -y install nginx
RUN pip3 install envtpl
```

```
FROM fedora:24
RUN dnf -y --setopt=tsflags=nodocs install nginx \
    && dnf clean all
RUN pip3 install envtpl \
    && rm -rf ~/.cache/*
```

378 MB ➡ 233.8 MB

SYSTEMD IN A CONTAINER

```
FROM fedora:24
RUN dnf -y --setopt=tsflags=nodocs install \
    nginx \
    uwsgi \
    uwsgi-plugin-python \
    && dnf clean all \
    && systemctl enable nginx \
    && systemctl enable uwsgi
ENV container=docker
STOPSIGNAL SIGRTMIN+3
RUN echo 'ForwardToConsole=yes' >> /etc/systemd/journald.conf
COPY uwsgi-app.ini /etc/uwsgi.d/
RUN chown uwsgi: /etc/uwsgi.d/uwsgi-app.ini
COPY nginx-app.conf /etc/nginx/nginx.conf
CMD [ "/usr/sbin/init" ]
```

```
docker run -dt -p 80:80 -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
    --tmpfs /run --tmpfs /tmp --name my-python-app \
    my-python-image
```

```
dnf install oci-systemd-hook
```

```
docker run -dt -p 80:80 --name my-python-app \  
my-python-image
```

READ-ONLY CONTAINER

```
docker run -d -p 8080:80 --read-only \  
--tmpfs /var/cache/nginx --tmpfs /run \  
--name my-nginx nginx
```

CONFIGURATION

- Environment variables
- Bind mount config dir
- Etc!

TEMPLATE + ENV VARS = CONFIG FILE

- sed
- envsubst < /my/template > /etc/config/file
 - bash variables
- envtpl < /my/template > /etc/config/file
 - jinja2
 - github.com/andreasjansson/envtpl

```
RUN pip3 install envtpl && rm -rf ~/.cache/*
```

ENVTPL TEMPLATE EXAMPLE

```
{% for key, value in environment('MY_APP_') %}{{ key }}={{ value }}  
{% endfor %}
```

INIT + CONFIG + EXEC = SHELL WRAPPER

```
COPY docker-cmd.sh /init  
CMD [ "/init" ]
```

```
: "${MY_APP_DB_HOST:='mysql'}"  
: "${MY_APP_DB_PORT:='3306'}"  
...  
envtpl < /my/template > /etc/config/file  
...  
exec /usr/sbin/init
```


MYSQL CONFIG AND INIT EXAMPLE

```
: "${MY_APP_mysql_host:=mysql}"
: "${MY_APP_mysql_port:=3306}"
: "${MY_APP_mysql_user:=${MYSQL_ENV_MYSQL_USER:-root}}"
if [ "${MY_APP_mysql_user}" = 'root' ]; then
: "${MY_APP_mysql_password:=${MYSQL_ENV_MYSQL_ROOT_PASSWORD}}"
fi
: "${MY_APP_mysql_password:=${MYSQL_ENV_MYSQL_PASSWORD:-myapp}}"
: "${MY_APP_mysql_dbname:=${MYSQL_ENV_MYSQL_DATABASE:-myapp}}"

MYSQL_COMMAND="mysql -h ${MY_APP_mysql_host} \
-P ${MY_APP_mysql_port} -u ${MY_APP_mysql_user} \
-p${MY_APP_mysql_password}"
```

Auto-config will work only when linking containers on default bridge

...

```
until $MYSQL_COMMAND -e ';' ; do
    >&2 echo 'MySQL is unavailable - sleeping'
    sleep 1
done

$MYSQL_COMMAND -e \
    "CREATE DATABASE IF NOT EXISTS ${MY_APP_mysql_dbname}"

MYSQL_CHECK_IF_HAS_TABLE="SELECT COUNT(DISTINCT table_name) FROM \
    information_schema.columns WHERE table_schema = \
    '${MY_APP_mysql_dbname}';"
MYSQL_NUM_TABLE=$( $MYSQL_COMMAND --batch --skip-column-names \
    -e "$MYSQL_CHECK_IF_HAS_TABLE" )
if [ "$MYSQL_NUM_TABLE" -eq 0 ]; then
    $MYSQL_COMMAND -D "$MY_APP_mysql_dbname" < /mysql/schema.sql
fi
```

SECURITY

- selinux
 - who can talk to who
- seccomp
 - what can be said

SELINUX

-v /host/path:/cont/path(:Z|:z)

--security-opt label:(user|role|type|level|disable):VALUE

SECCOMP

--security-opt seccomp:(/path/profile.json|unconfined)

```
{
  "defaultAction": "SCMP_ACT_ALLOW",
  "syscalls": [
    {
      "name": "getcwd",
      "action": "SCMP_ACT_ERRNO"
    }
  ]
}
```

github.com/pschiffe/docker-pdns

github.com/pschiffe/docker-borg

docs.ansible.com/ansible/docker_container_module.html

ansible.com/ansible-container

kubernetes.io

openshift.org



Feedback: schiffer.typeform.com/to/Su2TwF

github.com/pschiffe

linkedin.com/in/peterschiffer